

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A method for dynamically allocating data buffers to a data structure, comprising the computer-implemented steps of:

assigning a logging thread to said data structure, wherein said logging thread is

configured to insert free data buffers into said data structure;

monitoring a number of buffers that are moved into a full buffer list from the data structure;

determining, based on the number of buffers, throughput demands that are

currently required for a web site domain that is associated with the data structure;

based on the throughput demands that are currently required for the web site domain, changing a minimum number of data buffers that are to be consistently linked to the data structure; and

if a number of data buffers that are currently linked to the data structure is less than the minimum number, then performing steps comprising:

identifying one or more free buffers that are within a free buffer pool; and

linking said one or more free data buffers into said data structure;

wherein said one or more free data buffers are stored within a memory of a computer system.

2. (original) The method of Claim 1, further comprising the steps of:

receiving requests for content that is associated with a web site domain;

generating log data based on the requests; and

writing said log data in one or more data buffers associated with said data structure.

3. (previously presented) The method of Claim 1, wherein the step of linking said one or more free data buffers into said data structure includes linking said one or more free data buffers into said data structure into which one or more other data buffers already are linked.
4. (original) The method of Claim 1, further comprising the steps of:  
determining that a particular data buffer should be removed from said data structure;  
unlinking said particular data buffer from said data structure; and  
inserting said particular data buffer into a ready-to-write buffer list.
5. (original) The method of Claim 4, wherein the step of determining that a particular data buffer should be removed comprises the step of detecting that said particular data buffer is full.
6. (original) The method of Claim 4, wherein the step of determining that a particular data buffer should be removed comprises the step of detecting that said particular data buffer has not been removed from said data structure for a particular period of time.
7. (previously presented) The method of Claim 4, further comprising the steps of:  
removing said particular data buffer from said ready-to-write buffer list, wherein said ready-to-write buffer list is located within a first memory area;  
storing log data information in said particular data buffer to a second memory area, wherein said second memory area is distinct from said first memory area; and  
inserting said particular data buffer into said free buffer pool, wherein said free buffer pool maintains free data buffers that may be inserted into any one of

a plurality of data structures that are each associated with a particular web site domain.

8. (original) The method of Claim 7, wherein:  
the step of identifying one or more free buffers comprises the step of selecting one or more free buffers from said free buffer pool; and  
the step of linking said one or more free data buffers into said data structure comprises the steps of,  
identifying one or more entries in said data structure; and  
linking said one or more free data buffers into said one or more entries in said data structure.
9. (original) The method of Claim 4, wherein:  
said log data is generated based on request that are received for content associated with a particular web site domain; and  
said step of inserting said particular data buffer comprises the step of linking said particular data buffer into a queue that maintains only data buffers that contain log data associated with requests for said particular web site domain.
- 10-12. (canceled)
13. (Currently Amended) A computer-readable ~~storage~~ medium carrying one or more sequences of instructions for dynamically allocating data buffers to a data structure, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:  
assigning a logging thread to said data structure, wherein said logging thread is configured to insert free data buffers into said data structure;

monitoring a number of buffers that are moved into a full buffer list from the data structure;

determining, based on the number of buffers, throughput demands that are currently required for a web site domain that is associated with the data structure;

based on the throughput demands that are currently required for the web site domain, changing a minimum number of data buffers that are to be consistently linked to the data structure; and

if a number of data buffers that are currently linked to the data structure is less than the minimum number, then performing steps comprising:

identifying one or more free buffers that are within a free buffer pool; and

linking said one or more free data buffers into said data structure.

14. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 13, further comprising instructions for performing the steps of:  
receiving requests for content that is associated with a web site domain;  
generating log data based on the requests; and  
writing said log data in one or more data buffers associated with said data structure.
15. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 13, wherein the step of linking said one or more free data buffers into said data structure includes linking said one or more free data buffers into said data structure into which one or more other data buffers already are linked.
16. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 13, further comprising instructions for performing the steps of:

determining that a particular data buffer should be removed from said data structure;

unlinking said particular data buffer from said data structure; and

inserting said particular data buffer into a ready-to-write buffer list.

17. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 16, wherein the step of determining that a particular data buffer should be removed comprises the step of detecting that said particular data buffer is full.
18. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 16, wherein the step of determining that a particular data buffer should be removed comprises the step of detecting that said particular data buffer has not been removed from said data structure for a particular period of time.
19. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 16, further comprising instructions for performing the steps of:  
removing said particular data buffer from said ready-to-write buffer list, wherein  
said ready-to-write buffer list is located within a first memory area;  
storing log data information in said particular data buffer to a second memory area, wherein said second memory area is distinct from said first memory area; and  
inserting said particular data buffer into said free buffer pool, wherein said free buffer pool maintains free data buffers that may be inserted into any one of a plurality of data structures that are each associated with a particular web site domain.
20. (Currently Amended) The computer-readable ~~storage~~-medium of Claim 19, wherein:

the step of identifying one or more free buffers comprises the step of selecting one or more free buffers from said free buffer pool; and  
the step of linking said one or more free data buffers into said data structure comprises the steps of,  
identifying one or more entries in said data structure; and  
linking said one or more free data buffers into said one or more entries in said data structure.

21. (Currently Amended) The computer-readable ~~storage~~ medium of Claim 16, wherein:  
said log data is generated based on request that are received for content associated with a particular web site domain; and  
said step of inserting said particular data buffer comprises the step of linking said particular data buffer into a queue that maintains only data buffers that contain log data associated with requests for said particular web site domain.

22-24. (canceled)

25. (Previously Presented) A method for processing requests for content that is associated with different web site domains, the method comprising:  
receiving, at a web server, a first request for access to first content that is associated with a first web site domain of a plurality of web site domains;  
queueing the first request within a connection queue;  
assigning a first server thread of a plurality of server threads to service the first request;

in response to being assigned a task of servicing the first request, the first server thread determining to which web site domain of the plurality of web site domains the first request is related;

the first server thread loading first configuration data for the first web site domain in response to determining that the first request is related to the first web site domain, wherein, by loading the first configuration data, the first server thread is temporarily configured as a server thread that is dedicated to servicing requests for content that is available within the first web site domain;

the first server thread generating first log data based on the first content, wherein the first log data includes information that identifies a first web page that was requested within the first web site domain;

the first server thread selecting, from among a plurality of buffer files, a first buffer file that is associated with the first web site domain, wherein the first buffer file contains buffers that are to be used for buffering log data that is associated with the first web site domain;

the first server thread selecting a first buffer from among a plurality of buffers within the first buffer file;

the first server thread writing the first log data into the first buffer;

a logging thread selecting, from among a plurality of physical log files, a first physical log file that is associated with the first buffer file, wherein each physical log file of the plurality of physical log files is associated with a separate buffer file of the plurality of buffer files;

a logging thread storing, into the first physical log file, information contained within the first buffer;

receiving, at the web server, a second request for access to second content that is associated with a second web site domain of a plurality of web site

domains, wherein the second web site domain is separate from the first web site domain;

queueing the second request within the connection queue;

assigning a second server thread of the plurality of server threads to service the second request, wherein the second server thread is separate from and executes concurrently with the first server thread;

in response to being assigned a task of servicing the second request, the second server thread determining to which web site domain of the plurality of web site domains the second request is related;

the second server thread loading second configuration data for the second web site domain in response to determining that the second request is related to the second web site domain, wherein, by loading the second configuration data, the second server thread is temporarily configured as a server thread that is dedicated to servicing requests for content that is available within the second web site domain, wherein the second configuration data differs from the first configuration data;

the second server thread generating second log data based on the second content, wherein the second log data includes information that identifies a second web page that was requested within the second web site domain;

the second server thread selecting, from among the plurality of buffer files, a second buffer file that is associated with the second web site domain, wherein the second buffer file contains buffers that are to be used for buffering log data that is associated with the second web site domain, wherein the second buffer file differs from the first buffer file;

the second server thread selecting a second buffer from among a plurality of buffers within the second buffer file;

the second server thread writing the second log data into the second buffer;



a logging thread selecting, from among the plurality of physical log files, a second physical log file that is associated with the second buffer file, wherein the second physical log file is separate from the first physical log file; and  
a logging thread storing, into the second physical log file, information contained within the second buffer;  
wherein the first buffer is stored within a memory of a computer system.

26. (previously presented) The method of Claim 25, further comprising:

a logging thread determining whether to remove the first buffer from a buffer array that is contained in the first buffer file, wherein determining whether to remove the first buffer is based on at least one of (a) an amount of free space that is left within the first buffer and (b) an amount of time that the first buffer has been linked within the buffer array;  
in response to a logging thread determining that the first buffer should be removed from the buffer array, a logging thread moving the first buffer from the buffer array to a ready-to-write buffer list that is contained in the first buffer file.

27. (previously presented) The method of Claim 25, further comprising:

a logging thread determining whether a current time of day is a specified time of day;  
in response to a logging thread determining that the current time of day is the specified time of day, a logging thread determining whether the first buffer contains log data; and  
in response to a logging thread determining that the first buffer contains log data as of the specified time of day, a logging thread moving the first buffer from a buffer array, which is contained in the first buffer file, to a ready-to-write buffer list that is contained in the first buffer file.

28. (previously presented) The method of Claim 1, wherein linking said one or more free data buffers into said data structure comprises linking multiple data buffers into said data structure.
29. (Currently Amended) The computer-readable ~~storage~~ medium of Claim 13, wherein linking said one or more free data buffers into said data structure comprises linking multiple data buffers into said data structure.